# Navigation Control of an Automated Guided Underwater Robot using Neural Network Technique

A project report submitted in partial fulfillment for the degree of

**Bachelor of Technology**

In

**Mechanical Engineering**

by


**Prabhu Datta Nepak,**
Roll No-109ME0396,

**Sunit Nayak,**
Roll No-109ME0395

And

**Sambit Kumar Pradhan,**
Roll No-109ME0175


Under the Supervision of


**Prof. Dayal Ramakrushna Parhi**

Department of Mechanical Engineering
National Institute of Technology, Rourkela


**National Institute of Technology, Rourkela**

## National Institute of Technology Rourkela

# CERTIFICATE

This is to certify that the work in this thesis entitled **"Navigation Control of Automated Guided Underwater Robot using Neural Network Technique"** by **Prabhu Datta Nepak, Sunit Nayak and Sambit Kumar Pradhan**, has been carried out under my supervision in partial fulfillment of the requirements for the degree of **Bachelor of Technology** in **Mechanical Engineering** during session 2012 – 2013 in the **Department of Mechanical Engineering, National Institute of Technology, Rourkela.**

To the best of my knowledge, this work has not been submitted to any other University/Institute for the award of any degree or diploma.

**Dr. Dayal R. Parhi**

Department of Mechanical Engineering

National Institute of Technology
Rourkela

# ACKNOWLEDGEMENT

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide Prof. **Dayal R. Parhi**, Department of Mechanical Engineering, National Institute of Technology, Rourkela, in bringing this report to a successful completion. We consider ourselves extremely fortunate to have had the opportunity to work under the guidance of such a knowledgeable, dynamic and generous personality.

We are also thankful to Prof **K P Maity**, Head of Department, Department of Mechanical Engineering; National Institute of Technology Rourkela; for his constant support and encouragement.

Finally, we extend our sincere thanks to **Mrs S. Kundu** for her support throughout the project work.

| | | |
|---|---|---|
| Prabhu Datta Nepak | Sunit Nayak | Sambit Ku.Pradhan |
| Roll No-109ME0396 | Roll No-109ME0395 | Roll No-109ME0175 |

# CONTENTS

**Abstract**.......................................................................................................................1
**Chapter 1 Introduction**.................................................................................................2
**Chapter 2 Literature Review** .........................................................................................5
**Chapter 3 Analysis of Neural Network for underwater robot** ......................................15
3.1 Introduction to Neural Networks ......................................................................16
3.2 A Simple Neural Network Model ......................................................................18
3.3 Feed forward Neural networks...........................................................................19
3.4 Back Propagation ..............................................................................................20
3.5 Modeling of the Neural Network for the underwater robot .............................20
**Chapter 4 Algorithm for programming in MATLAB**.........................................................27
**Chapter 5 Results and Discussion** ...............................................................................31
**Chapter 7 Conclusion and Scope for Future Work**............................................................36
**References**.....................................................................................................................38

# List of Figures

Figure 1 Biological Neural Network ........................................................................ 17

Figure 2 Artificial Neural Network ........................................................................ 18

Figure 3 Feed Forward Neural Network.................................................................. 19

Figure 4 Neural Network model used for the underwater robot....................................... 24

Figure 5 Neural Network Training in MATLAB ............................................................ 33

Figure 6 Variation of Mean Square Error with no of epochs ........................................... 33

Figure 7 Variation of Gradient and Learning Rate........................................................ 34

Figure 8 Regression plot of the neural network .............................................................. 34

Figure 9 Simulation of the underwater robot with obstacles............................................35

# List of Tables

Table 1 Different situations used in the neural network.................................................. 27

# ABSTARCT

In recent years, under water robots play an important role in various under water operations. There is an increase in research in this area because of the application of autonomous underwater robots in several issues like exploring under water environment and resource, doing scientific and military tasks under water. We need good maneuvering capabilities and a well precision for moving in a specified track in these applications. However, control of these under water bots become very difficult due to the highly non-linear and dynamic characteristics of the underwater world. The logical answer to this problem is the application of non-linear controllers. As neural networks (NNs) are characterized by flexibility and an aptitude for dealing with non-linear problems, they are envisaged to be beneficial when used on underwater robots. In this research our artificial intelligence system is based on neural network model for navigation of an Automated Underwater robot in unpredictable and imprecise environment. Thus the back propagation algorithm has been used for the steering analysis of the underwater robot when it is encountered by a left, right and front as well as top obstacle. After training the neural network the neural network pattern was used in the controller of the underwater robot. The simulation of underwater robot under various obstacle conditions are shown using MATLAB.

CHAPTER 1

# INTRODUCTION

# Introduction

Underwater Robots are beneficial and substantial tools to achieve various actions and have increasing applications in discovering underwater environments. These applications include good maneuvering capabilities and a high accuracy path tracking. Due to the uncertainties in underwater environment and nonlinearity of the dynamic behavior a suitable control form of the underwater robot is very difficult. Various parameters are needed to monitor and control by the operator during the implementation of certain task with the underwater robot. Therefore, research in the area of monitoring and controlling the underwater robots become an important challenge for the engineers and have been considered by many researchers. If some parameters like position and direction of motion of the robot could be controlled automatically by a control system the underwater robot can be enormously facilitated. The control system is one of the most important component of an underwater robot, and its features play an important role when one have to choose a robot for a specific mission. These robots are being substituted by the human divers for completing different tasks that over risks the human life.

Some linearization theory for steering an underwater robot controllers were presented in the last decade. Steps have been taken to obtain the tracking control of the accurate path as the dynamics is uncertain. The objective of the underwater robot is to build a platform which is capable of successfully navigating in the underwater environments without any human support. The result of the techniques which navigation methods are based upon; differ in relation to the computational time and resources available to the robot, the dynamic and static memory available to it, the accuracy and efficiency of performance expected from the robot. They are usually classified by the characteristics of the underwater environment in which the robot is to navigate.

One of some approaches for the navigation control of underwater robot is Artificial Neural-Networks (ANN) for reactive control. The benefit of this approach is the learning capability of the neural network. Most underwater robot show non-linear characteristics, linear control cannot be used for all circumstances. The reasonable answer to this problem is the application of non-linear controllers. As neural networks (NNs) are characterized by flexibility and an aptitude for dealing with non-linear problems, they are envisaged to be beneficial when used on underwater robots.

The neural network is a method of understanding theories which leads to the way in which human brain comprehends primary functions. Neural Network is used to construct machines which perform complicated task such as optimization, learning, adaptation, and generalization. Since the navigation problem constitutes of recognition, learning, decision-making and action. For navigating the underwater robot the algorithm used is neural network. The learning and adaptation capabilities can be improved by using Neural Network in the environments where the data is qualitative, inaccurate, and vague or incomplete. Most underwater show non-linear behavior, linear control cannot be used for all circumstances. The logical answer to this problem is the application of non-linear controllers. As neural networks (NNs) are characterized by flexibility and an aptitude for dealing with non-linear problems, they are envisaged to be beneficial when used on underwater robots.

CHAPTER 2

# LITERATURE REVIEW

# Literature Review

Chen, et al. [1] proposed robust adaptive control for the underwater robot in the presence of parametric nonlinearity, external disruptions, input dead zone and saturation. They introduced Backstepping control of the system dynamics and hence develop full state feedback navigation control. They presented robust adaptive tracking control for under water robots using Back stepping control, parameter adaption and variable structure to handle the nonlinearity, impregnation and dead zone characteristics of the actuator used in the robot. The dead-zone and saturation of the actuator are considered during the control design. Since neural network have inherent approximation capabilities they use Neural Network as an approximator for system parametric uncertainties. They proposed the full state feedback for the underwater robot with backstepping method and neural networks to challenge saturation and deadzone nonlinearities of actuators, the sliding mode tracking control. In this research, two auxiliary design variables have been introduced to design backstepping control to shorten the derivation computation of the virtual control law and to tackle the input nonlinearity. In conclusion, they have presented reports of simulation studies to demonstrate the efficiency of the discussed tracking control.

Parhi and Singh [2] did a research in which about the responsive control of an autonomous robot which is able to travel safely in a crowded unknown environment and to reach predefined target by dodging static as well as dynamic obstacle. The neuro controller is designed using a four layer neural network to solve the time and path optimization problem of the robot. The used the of left, right, and front obstacle distance to its locations as the inputs and target angle between the robot and specified target as output of the neural network. The input and outputs are sensed by an array sensor. Different cognitive tasks are carried by the neuro controller such as adaptation, learning, optimization and generalization. The neural network is trained by Back Propagation algorithm. They also analyzed kinematical modeling of mobile robots as well as the design of control systems for the autonomous motion of the robot in this research. A real practical setup had been done to show the performance of training of the networks.

Guo et al. [3] carried out a research for path-planning of an underwater robot in 3D world by creating a novel spiral particle pathway searching approach. A multi-microbot system was proposed by them which have a mother submarine and various micro robots with Ionic Conducting Polymer film Actuators. The environmental model was set up by considering various factors of underwater space for robot path planning in 3D space. They have taken the spheres as objects and a point as the robot and separate the 3D space into several parts. Considering the concept of the Grid Method a Spiral Particle Pathway Searching Approach is established and is used to explore for obstacles in the passageway in the plane of parallel subspace. The best path to reach the target is searched using MATLAB. DPSO is utilized to get the optimize result.

Ishii and Ura [4] carried out a research of an adaptive neural network controller system for an underwater robot. There proposed system consists of two systems. First one is a real-world part and second one is an imaginary-world part. The real-world part is for the feedback control system for the robot. The imaginary-world part is for the model of robot and the controller are adjusted constantly in order to deal with deviations of dynamic properties caused by disturbances and so on. To suit the parallel processing ability with the computer system the proposed system is designed. They examined the adaptability of the controller system by heading-keeping and path-following experiments in circumstances where capricious disturbances are applied to the robot.

Zhang and Chu [5] two researcher of Harbin Engineering University researched on the path tracking control problem of underwater robot. They used local recurrent neural networks for underwater robot for adaptive sliding control of underwater robot. As an precise thrust modeling is very tough to establish for underwater robot. So they use a control voltage of thruster to design the input of the system by the controller. Taylor’s polynomial is used to transform the form of path tracking error system of underwater robot to the form of a fine nonlinear system, whose input is the control voltage of thruster. After that, according to the principle of sliding mode control, using the recurrent neural network they estimate the unknown item of affine system online an adaptive sliding mode control is proposed.

Another paper by the Vena et al. [6] shows the research of use neural networks in the identification of models for underwater vehicles. To account for unmodelled phenomena neural network in parallel is used. Rather than using the above method details of the various parts of the model is used to apply neural networks for the uncertain parts. The damping of an underwater vehicle can be recognized using neural networks. Then the neural network based model performance is demonstrated in simulations using the neural networks in a feed forward controller.

There are some researches where a neuro-fuzzy logic is used for the motion control of the robots. Kim and Yuh [7] of Autonomous Systems Laboratory, Department of Mechanical Engineering researched for using neuro-fuzzy logic to autonomous underwater vehicles. In that research they used a neuro-fuzzy controller for autonomous underwater vehicles (AUVs) where the dynamics are highly nonlinear, and vary with time. Fuzzy membership function based on neural networks is used by the neuro fuzzy controller. Advantages of fuzzy and neural networks are implementing human experience with fuzzy logics, and approximating learning capability with neural networks. There is no need of any information about the system, learning offline, or human interference to adjust parameters in fuzzy membership function-based neural networks (FMFNN) controller like other conventional approaches. Using inner loop learning and vehicle system derivatives, online learning of the FMFNN controller can be achieved.

Lianzhi and Weichong [8] researched on using Cascade Hopfield Neural Network Model for motion control of a robot. As the robot moving modes in one cycle, the cascade Hopfield neural network model with three neural nodes was set up. After that the weight factors and thresholds of the networks were designed. The results show the adaptability of cascade Hopfield neural network controller for the orderly continuous moving process of any robot.

Joshi and Zaveri [9] studied neural network that can be used for navigation of robot because of their learning ability to humans. This can be used to develop navigation strategies for autonomous robot. They developed neural network based systems for mobile robot navigation. The mentioned systems transform sensors input to obtain wheel velocities. For optimal training of neural network novel algorithm is used. In order to ensure the efficacy of proposed system; proposed neural system's performance is compared to other neural and fuzzy based approaches.

There is another research by Lin and Xie et al. [10]. Their research is based on supervised neural network based motion control of a bionic under water robot. A supervised Neural Q_learning (SNQL) algorithm is used for controlling the robot. This algorithm is based on conventional Q_learning algorithm which is a type of reinforcement algorithm, but has three remarkable distinctions: (1) A Feed forward neural network is used to approximate the

Q_function table; (2) For speed up the learning and stability system a learning sample database is adopted; (3) introducing a supervised control in the earlier stage of learning for safety and to speed up learning again. Experiments are carried out with SNQL algorithm and the results shows that the SNQL algorithm is more effective than pure neural Q_learning or supervised control.

Parhi et al. [11] carried out a research on intelligent path planning of multiple mobile robots. Cooperative behavior can be attained using several mobile robots, which can communicate online via inter-communication among themselves. For steering mobile robots which are used in non-linear or unknown environment neuro-fuzzy techniques are observed by them. The ultimate aims of the robots are to reach some pre-defined goals. Based upon a orientation, direction of the robot; distances between the robots and obstacles; and distances between the robots and targets different types of rules are taken and refined later to find the steering angle. The control system combines a repelling influence related to the distance between robots and nearby obstacles and with an attracting influence between the robots and targets. To find the steering angle of the robots a hybrid rule-based-neuro-fuzzy technique is analyzed. Results show that the navigation performance in complex and unknown environments can be improved by the proposed rule-based-neuro-fuzzy technique compared to this simple rule-based technique.

Kundu et al. [12] carried out a research based on a new prototype of intelligent navigation system for mobile robot that has been augmented with some common features like: criteria for optimal performance and ways to optimize design, structure and control of robot. It would be valuable to immaculately combine robust learning capabilities of artificial neural networks with a high level of knowledge interpretability provided by fuzzy-logic with the growing need for the deployment of intelligent and highly autonomous systems. In view of sensor-rich system with real time constraints by adaptive learning, rule extraction and insertion, and neural/fuzzy reasoning fuzzy-neural network is able to build comprehensive knowledge bases. They simulated the technique and also compared with other simulation studies by previous researcher. An analysis has been done in real experiment setup for the training of back propagation algorithm and its navigational performances analysis. As experimental result matches well with the simulation result, the practicality of method is verified.

Kodogiannis et al. [13] carried out a research for navigation control using neural network and to identify the system of Underwater Robotic Vehicles, consists of a system containing severe unconformities. NNs models are developed and then incorporated into predictive control strategies which are evaluated on-line. They compare the results both the modelling and control of the system including hybrid control strategies which combine neural predictive with conventional three term controllers.

Fernandes et al. [14] carried out a research to use robotic vehicles for underwater exploration. Their work describes the development of a dynamic positioning system for remotely operated underwater robot. The adopted approach is developed using Lyapunov Stability Theory and enhanced by a neural network based algorithm for uncertainty and disturbance compensation. They evaluated the performance of the proposed control scheme by means of numerical simulations.

As there has been an increase in the necessity of underwater bots for various application, Bagheri and Moghaddam[15] two researcher of Department of Mechanical Engineering of the University of Guilan have done some research on Simulation and tracking control of underwater remotely operated vehicle based on neural-network strategy. They designed and fabricated a remotely operated vehicle for exploring underwater with special application for monitoring and studying fish behavior in the Caspian Sea. In that research they use a sliding-mode neural-network scalar (SMNNS) control system to track the control of the path of the underwater vehicle for achieving a high-precision position control. A neural-network controller is used in SMNNS for an equivalent control law in the sliding-mode control, and a robust controller and also a scalar controller are designed to curb the system dynamics on the sliding surface for guaranteeing the asymptotic stability property and achieving high-accuracy position control.

Another research of same University by Bagheri et al. [16] of Department of Mechanical Engineering of the University of Guilan which was on tracking performance control of a cable communicated underwater vehicle using adaptive neural network controllers. The design, dynamic modeling and control of a fabricated underwater remotely operated vehicle are shown in that paper. Dynamic model of the vehicle has four degrees of freedom and they represent the dynamic effects of the towed cable which is used for dynamic simulation and control design. They developed a nonlinear adaptive neural network controller and simulated it. The adaptive controllers follow Multilayer and radial basis function neural networks. The performance of the vehicle with neural network controllers is compared with a PD controller.

The substantial improvement is observed for tracking performance of the vehicle in all controllable degrees of freedom.

Chatchanayuenyong and Parnichkun [17] of School of Engineering and Technology, Asian Institute of Technology published a paper which shows the research of using a neural network based-time optimal sliding mode control for an autonomous underwater robot. The SMC follows the Pontryagin’s time optimal control principle, in which the solution is obtained by neural network approach. They implement it on a six-degree-of-freedom autonomous underwater robot developed. The proposed controller is analyzed for the performance and compare with other SMCs and conventional linear systems. They analyzed and presented the comparison detail results such as controller performance and error phase portrait. Such evaluations ensure the application success and verify it as a real time-optimal controller.

Fujii and Ura [18], researched for the application of neural network controlling the underwater robots. They designed an adaptive controller which is based on neural network which is called self-organizing neural-net control system (SONCS). A Feedback controller based on back propagation algorithm is used. Kazuo Ishii, Tamaki Ura researched and published a paper on this. They proposed SONCS system for underwater robot.

Li and Lee [19] researched for diving control of a underwater robot using a neural network adaptive controller. Since the dynamics of underwater vehicles are highly nonlinear and their operating environments are hard to predict accurately a priori. Due to this reason, the autonomous diving equation of an Autonomous underwater Vehicle (AUV), a certain simplification of 6 DOF nonlinear equations of URVs, may comprise various unbound uncertainties. In this research, they recommend a robust neural network adaptive control scheme for autonomous diving control of an AUV in order to deal with these unbound uncertainties. Although they still gratify certain growth conditions characterized by ‘bounding functions’ unstructured uncertainties are supposed to be unbounded. They derived all adaptation laws for unknown bounds of uncertainties the Lypunov-based method as well as the update laws of the networks’ weights values. They did not approximate the unknown control gain functions directly using neural networks and therefore can evade the possible controller singularity problem. They also included simulation studies to exemplify the usefulness of the presented control scheme. They also discussed some practical features of the control law. Additional surveys on how to construct the basis function vectors and somewhat simply the structure of the stabilizing functions may be needed in the future practical applications.

Many of the Adaptive PD controllers use the theory of neural network for the nonlinear movements of the robots. Hoang and Kreuzer [20] of the Hamburg University of Technology, Mechanics and Ocean Engineering has researched in this area to position a remotely operated vehicle under water. The requirement for high accuracy in dynamic positioning of remotely operated vehicles especially when tasks close to underwater have to be performed needs high precision of sensor systems.

Lee and Choi [21] present a simple control technique using a multilayer neural network with the backpropagation learning methodology. This multilayer neural network was acting as a compensator for the traditional sliding mode controller to enhance the control performance when initial presumption of uncertainty limits of system parameters are not conclusive. The suggested controller is applied to precisely control a robot working under the water which has large uncertainties such as the drag force, buoyancy, currents, wave effects, and the additional mass moment of inertia. Simulation results from computer showed that the suggested control scheme gives an yielding path way to adapt with the unexpected situations encountered under water.

Ouarda [22] carried out a research on neural network based navigation for intelligent autonomous mobile robots have been discussed. Neural Networks deal with cognitive tasks such as learning, adaptation generalization and they are well appropriate when knowledge based systems are involved. The adaptation is largely related to the learning capacity since the network is able to take into account and respond to new constraints and data related to the external environments. Just as human being, a neural network relies on previously solved examples to build a system of “neurons” that makes new decisions, classification and forecasts. Networks of neurons can achieve complex classification based on the elementary capability of each neuron to distinguish classes its activation function. In designing a Neural Networks navigation approach, the ability of learning must provide robots with capacities to successfully navigate in the environments like our proposed maze environment. Also, robots must learn during the navigation process, build a map representing the knowledge from sensors, update this one and use it for intelligently planning and controlling the navigation. The simulation results display the ability of the neural networks based approach providing autonomous mobile robots with capability to intelligently navigate in several environments.

Zhang et al. [23]. An artificial bionic neural network to control fish-robot locomotion is presented. The fish-robot is modeled as a multi-joint dynamic system with parallel connections and composed of several motors. They analyzed the principle of the central pattern generators (CPGs) governing the locomotion of fish. An artificial neural network which has much comparability with the biological CPGs is designed to control the fish-robot. Their experimental results of startup, stop, forward swimming and backward swimming show its validity and efficiency.

Ayrulu and Barshan [24] have done a research on neural networks and investigated the processing of sonar signals for the robustly differentiating commonly encountered features that are found in indoor robot environments. This distinction of features is significant for a range of applications for intelligent systems. They have present the processing of various representations of amplitude and times of flight measurement patterns. These patterns are in turn obtained from a real sonar system. In this research, modular and non-modular neural networks have been trained. Back-propagation had been used to train the neural network with the use of generating-shrinking algorithms for to develop a learning process for identification of relations between parameters for the target primitives. Better generalization and interpolation capability as well as a faster convergence rate were shown by the networks which are trained with this mentioned generating-shrinking algorithm. They analyzed the neural network and came to know they can be used to reach different targets with single sensor only. Further, it results in a higher correct differentiation percentage than what was achieved by using methods that existed previously that depended on multiple sensor nodes. A pair of transducers which are defined by single sensor node with constant space can move and scan the destination for collecting information. They detected if the sensing nodes are decreased then the performance of other procedures is decreased. As it demonstrates that sonar signals can give information sufficient for differentiating all types of targets, the neural network approach is successful. They showed some examples shown of the system control based on acoustic signal detection and identification, obstacle avoidance, map-building, target-tracking, navigation etc. for autonomous mobile robots as well as other intelligent systems.

Soylu et al. [25] in this paper has considered an Automatic Guided Vehicle routing problem. The problem is to find the shortest route through which it can carry out multiple number of pick and deliver for a single, free ranging AGV. An Artificial Neural Network algorithm has been introduced based on Kohonen's self-organizing maps. The performance of the algorithm is tested for various P and D tasks and parameter settings. The results obtained from nearest neighbor rule and optimal solution rule are compared with these results. Result for computation time in the artificial neural network algorithm has been obtained in this research.

Orille et al. [26] carried out a research to develop a multi-layer network similar to that of a traditional switching look up table method. They used it for an induction motor Direct Torque Control (DTC) that results a switching pattern which is optimal. For the development of the system the MATLAB/SIMULINK program has been used. MATLAB has been used for the simulation. An effective neural network configuration is designed and tested by appropriately choosing the neural that represents the switching look-up table. To find out the stability and reliability of the system the neural network representing the switching look-up table is tested as part of the Direct Torque Control (DTC). To evaluate the performance of the neural network in its simulation for the DTC the simulation result obtained by the switching look-up table is used as a reference.

Scarselli and Tsoi [27] have produced works on approximation methods by feedforward neural networks. Problems seen in computational aspects of the method have been focused on this paper. Developing a feedforward neural network that can obtain a predetermined degree of approximation has been discussed in this paper. To obtain the required accuracy of approximation the number of hidden layers is determined. To understand the problem of universal approximation by using feedforward neural networks a unifying structure was proposed. For training two algorithms are taken which determine the corresponding weights of the given inputs in the feed forward neural network where the sigmoid function is used as the activation function. One of the common problems which plague several neural network training algorithms is escaping from the local minima and the trained algorithm is successful in it.

# CHAPTER 3

# ANALYSIS OF NEURAL NETWORK FOR UNDERWATER ROBOT

## 3.1 Introduction to Neural Networks

An Artificial Neural Network (ANN) is a model for information processing which has been developed from a correlation with biological nervous system and it is same as how the brain analyzes information. It is similar to the axons and dendrites that are present in the nervous system and is a simulation of the nervous system that contains a set of neuron units which are interrelated with each other though axon connections. The processing of data is the basic idea for this neural network model. These interconnected neurons help in solving real life problem.

An Artificial neural network should be trained so that it will work in a practical way. The network learns just like the human beings. An ANN is generally used for a specific application after training. The work done by the network varies from decision making, processing and optimization of image, optimization. Neural networks are also utilized for other applications like data mining application, classification of different items, descriptive modeling, approximation of functions, clustering, predicting of different series etc. The real neural networks present in brains can be understood better using artificial neural networks. There is no need to make a real biological neural model system to solve artificial intelligence situations or problems. Artificial Neural Networks also utilize the feature of adjusting the weight between the neurons present in the network.

The brain of human being contains set of neurons which are highly interconnected by axons and dendrites. Activation signal is transmitted between neurons by which information travels. This is how decision making is done in human brain.

The artificial neural network is like a mathematical version of the real neural network shown above. Artificial Neural network consists of neurons which are interconnected to communicate by using activation signals. The ANN can be used to approximate a function of many inputs and outputs that is applied to a particular use. The analogies between real neural network and the ANN have been given below.

The human brain is used for the continuous processing of a large number of information in highly changing situations. The brain is able to achieve such difficult tasks by using multiple processing of different input and output elements. Artificial Neural Networks utilizes the same technique to the compute the output for simulation.

The neuron contains three different parts; these are the neuron cell body or cyton, axon which transmits the neuron's output to the different dendrites of different neurons and dendrites which are the connecting extensions from cyton for taking input.

Dendrites receive the input signal from different neurons. Synapses which are the connections between different neurons results in the transfer of signals between neurons. These are of different types and are computed depending upon other properties like the speed and information in a signal. It has been anticipated that the nervous system of human consists of more than 100 billion neurons.

These neurons transmit electrical signals via a thin, long connector known as the axon which divides into a many branches. These synapses are present at the ends of every branch. They convert this into electrical signal. The next neuron in the path receives the input that is huge as compared to its inhibitory input, it sends a electrical signal down to the axon. The network trains itself by changing the effectiveness of the synapses so as to manipulate effect of neurons on the other neurons. The interactions are different for different types of network algorithim.

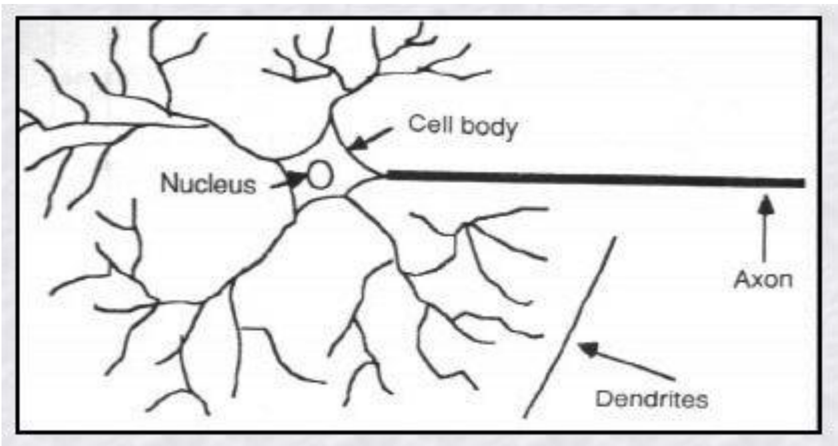


Figure 1Biological Neural Network [28]

The input signals at the synapses of the neuron are brought together for analysis. The excitatory and inhibitory influences occurring on the neuron are thus added. The output is not a linear function of the inputs signals and the weight of the connection in the synapses can be effectively changed by learning. The neuron fibre sends information to other neurons when the excitatory influences become dominant via the outgoing synapses. Firing occurs when the combined signal strength becomes more than a threshold value. Generally activation function gives the value to the neuron in the network.

The human brain trains itself by changing the nature and strength of the synapses. ANN trains itself algorithmically to get better results. In an ANN the learning of the situation is simulated mathematically by updating the weights acting between the neurons similar to that of human brain.The biological learning of human brain is simulated mathematically in ANN by modifying the weights between the neurons.

## 3.2 A Simple Neural network model

In an Artificial Neural Network the weight corresponds to synapse from a real neural network. When the weight is less than zero it represents an inhibitory connection, and when it's positive it represents an excitatory connection. All inputs are added together and are also changed by their changing weights. The process is called as linear combination. The vallue of the output is restricted by an activation function.

The following figure describes the mathematical algorithm.

Consider an artificial neuron network with n inputs, namely $I_1$, $I_2$, ...,$I_n$.

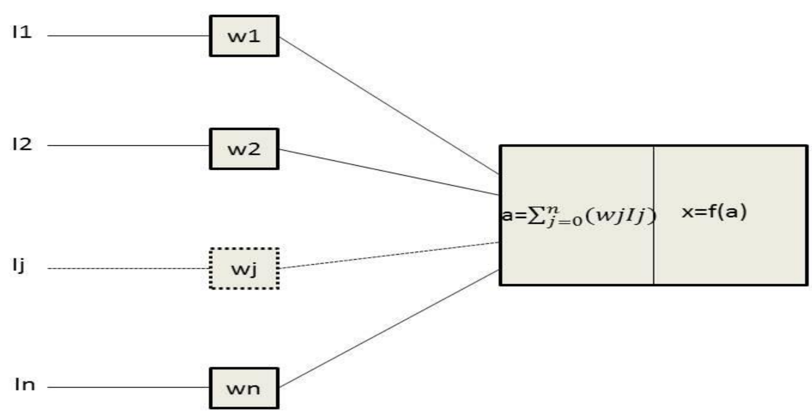


Figure 2 Artificial Neural Network

The lines connecting these inputs to other neurons are given some weights designeted by $w_1$, $w_2,\ldots,w_N$ respectively. For the particular graded potential the activation is determined by the formula: $w_j\ u_j$ . This model describes the interval activity of the neuron as

$$a=\sum_{j=0}^{n}(wj\ uj)$$

The weighted sum value _a' is the net input to the unit & the output of the activation function on the value of a, is the output of the neuron.

x = f (a )

The unit's activation function is the function f. In the simplest case, the unit's output is its net input and f is an identity function.

## 3.3 Feed forward networks:

In Feed forward neural networks the signals travel in only one direction i.e. from input to the output. These are differentiated by absence of feedback loops in the neural network. Therefore the output of one layer does not have any effect on the same layer. Feed-forward artificial neural network are forward networks that connects the inputs directly with the outputs. They are also called as bottom up neural network.

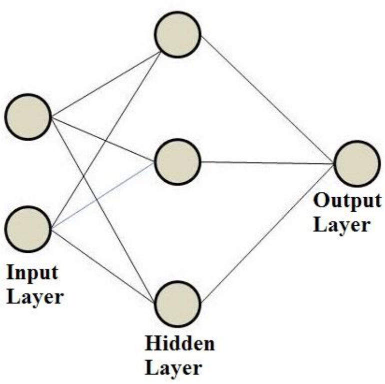


Figure 3-Feed Forward neural network

## 3.4 Back Propagation Algorithm

Back propagation is a learning method to train artificial networks and enforce the Delta rule. It is generally used for feed forward neural networks.

**Back propagation technique:**

1. A training set of data is provided to the neural network and the neural network‘s output is obtained.
2. The neural network output is compared with the desired output as given by the training set and the error is calculated at each output neuron.
3. For every neuron, the local error is calculated which gives the idea of how much lower or higher the output must be changed to be comparable with the desired output.
4. The weight of every neuron is changed to reduce the local error.
5. Blames for the local error are assigned to neurons which are present at the previous level, providing higher responsibility to neurons having higher weights.
6. Steps 3 to 5 are repeated for the neurons at the previous level by using the neuron’s blame as the error.

## 3.5 Modeling of the neural network for Underwater Robot

The steps required to train the neural network model used in the underwater robot is as follows-:

1. Collect Data-:

During training and during normal operation, the input patterns fed to the neural network comprise the following components[2]:

Y1 {1} = Left obstacle distance from the robot

Y2 {1} = Front obstacle distance from the robot

Y3 {1} = Right obstacle distance from the robot

Y4 {1} = Target bearing

And the final output

θactual=The change in steering angle

A set of data is to be collected for training of the neural network by taking all the various ways taking into consideration. Some of the

2. Create the network-:

Artificial neural networks consist of a set of densly interconnected processing units which are called neurons. These units transform signals in a non-linear way. Neural networks are non-parametric estimators which can fit smooth functions based on input-output examples. The neural network used is a five-layer perceptron. The chosen number of layers was found empirically to facilitate training. The input layer has four neurons, three for receiving the values of the distances from obstacles in front and to the left and right of the robot and one for the target bearing. If no target is detected, the input to the fourth neuron is set to 0. The output layer has a single neuron, which produces the steering angle to control the direction of movement of the robot. The first hidden layer has 8 neurons, the second hidden layer has 12 neurons and the third hidden layer has 6 neurons.

These numbers of hidden neurons were also found empirically. Figure 4 depicts the neural network with its input and output signals. The neural network is trained to navigate by presenting it with 60 patterns representing typical scenarios, which are depicted in Figure Table 1.

3. Configure the Network-:

The input neurons are[2]-:

Y1 {1} = Left obstacle distance from the robot ………………………(1)

Y2 {1} = Front obstacle distance from the robot……………………….(2)

Y3 {1} = Right obstacle distance from the robot……………………….(3)

Y4 {1} = Target bearing…………………………………………………(4)

These input values are distributed to the hidden neurons which generate outputs given by[2]

Yj{lay}= f (Vj{lay})…………….(5)

Where

$Vj\{lay\} = \Sigma Wji\{lay\}.Yi\{lay-1\}$……….(6)

lay = layer number (2 or 3)

j = label for jth neuron in hidden layer ‘lay’

i = label for ith neuron in hidden layer ‘lay-1’

Wji{lay} = weight of the connection from neuron i in layer ‘lay-1’to neuron j in layer ‘lay’

f(.) = activation function, chosen in this work as the hyperbolic tangent function :

$f(x) = (e^x - e^{-x})/(e^x + e^{-x})$

4. Initialize the weights and biases-:

Using all the collected data we can initialize the weights and biases. We can use MATLAB software for creating the Neural Network and solving the problem. After Creating the neural network and configuring it the weights should be initialized by proper values for further training of the neural network.

5. Train the Network-:

During training, the network output θactual may differ from the desired output θdesired as specified in the training pattern presented to the network. A measure of the performance of the network is the instantaneous sum-squared difference between θdesired and θactual for the set of presented training patterns[2]:

$Error = \frac{1}{2}\Sigma(\theta^2_{desired} - \theta^2_{actual})$……………….(8)

The error back propagation method is employed to train the network .This method requires the computation of local error gradients in order to determine appropriate weight corrections to reduce Err. For the output layer, the error gradient δ {4 } is[2]:

$\delta\{4\} = f'(V1\{4\})(\theta_{desired} - \theta_{actual})$……………..(9)

The local gradient for neurons in hidden layer {lay} is given by:

$\delta j\{lay\} = f'(Vj\{lay\})(\sum_k \delta k\{lay+1\}Wkj\{lay+1\})$…………….(10)

The synaptic weights are updated according to the following expressions:

$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}(t+1)$…………………..(11)

and $\Delta W_{ji}(t+1) = \alpha \Delta W_{ji}(t) + \eta\, \delta_j\{lay\}.Y_i\{lay-1\}$…………………(12)

$\alpha$=momentum coefficient (chosen empirically as 0.9 in this work)

$\eta$ = learning rate (chosen empirically as 0.05 in this work)

t = iteration number, each iteration consisting of the presentation

of a training pattern and correction of the weights.

The final output from the neural network is:

$\theta_{actual} = f(V_1\{4\})$………………………(13)

where

$V_1\{4\} = \sum_i Wli\{4\}Yi\{3\}$………………….(14)

It should be noted learning can take place continuously even during normal target seeking behavior. This enables the neural controller to adopt the changes in the robot's path while moving towards target. The proposed neural controller and kinematics gives steering angle from wheel velocities based on the environmental conditions.

6. Validate the Network-:

After training the network we have to validate it in the MATLAB so that it can be used by the robot for its navigation control.

7. Use the network-:

After that this network can be used by the network controller of the robot.

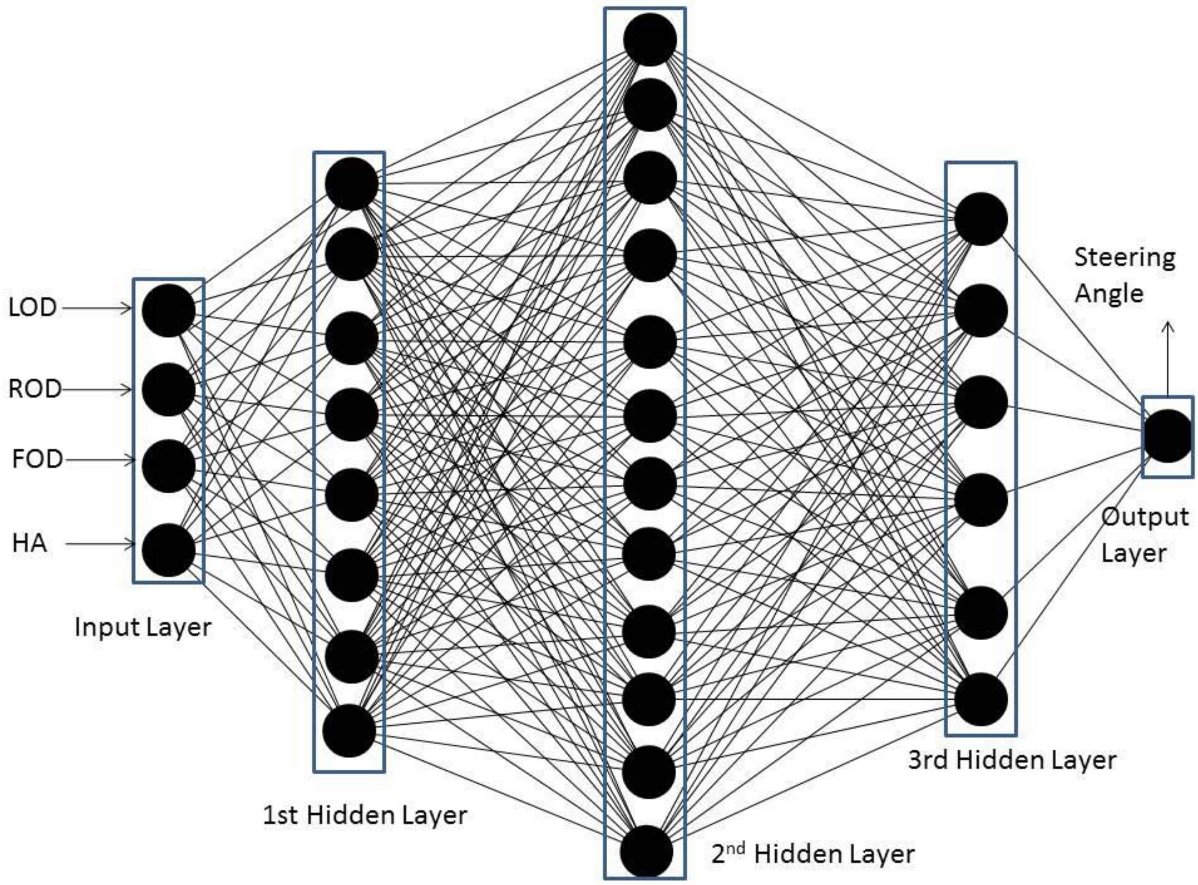


Figure 4-Neural network model used in the underwater robot

The different situations used in the neural network are shown below which are used in the neural network.

| LOD(in cm) | ROD(in cm) | FOD(in cm) | TA(in degree) | SA(in degree) |
|---|---|---|---|---|
| 60 | 10 | 10 | 12 | -7 |
| 60 | 15 | 10 | 12 | -7 |
| 60 | 25 | 10 | 12 | -7 |
| 60 | 20 | 15 | 15 | -7 |
| 60 | 25 | 15 | 15 | -7 |
| 60 | 30 | 15 | 15 | -7 |
| 60 | 40 | 10 | 20 | -7 |
| 50 | 40 | 10 | 20 | -7 |
| 55 | 40 | 12 | 20 | -7 |
| 60 | 35 | 12 | 25 | -9 |
| 60 | 35 | 12 | 25 | -9 |
| 25 | 35 | 25 | 25 | 9 |
| 20 | 35 | 25 | 25 | 9 |
| 25 | 35 | 30 | 25 | 9 |
| 25 | 35 | 30 | 12 | -9 |
| 10 | 60 | 30 | 12 | 7 |
| 10 | 60 | 30 | -12 | 7 |
| 20 | 60 | 20 | -12 | 7 |
| 25 | 60 | 20 | -12 | 7 |
| 25 | 60 | 8 | -15 | 9 |
| 25 | 60 | 8 | 20 | 9 |
| 30 | 60 | 8 | -15 | 9 |
| 60 | 60 | 28 | -15 | 8 |
| 60 | 60 | 28 | -25 | 8 |
| 25 | 25 | 28 | 30 | -7 |
| 22 | 25 | 22 | -25 | -8 |
| 25 | 20 | 15 | -25 | -8 |
| 15 | 30 | 25 | 25 | 8 |
| 10 | 25 | 25 | 25 | 8 |
| 10 | 25 | 30 | 30 | 8 |
| 20 | 25 | 10 | 20 | 10 |
| 30 | 25 | 30 | 20 | 10 |
| 20 | 25 | 30 | 15 | 4 |
| 20 | 25 | 15 | 30 | 14 |
| 30 | 25 | 25 | 30 | 14 |
| 30 | 40 | 20 | 15 | 6 |
| 30 | 25 | 25 | 15 | 6 |
| 25 | 30 | 25 | -15 | -6 |
| 40 | 30 | 20 | -15 | -6 |
| 25 | 30 | 25 | -30 | -14 |

| | | | | |
|---|---|---|---|---|
| 25 | 20 | 15 | -30 | -14 |
| 25 | 20 | 30 | -15 | 4 |
| 25 | 30 | 30 | -20 | -10 |
| 25 | 20 | 10 | -20 | -10 |
| 25 | 10 | 30 | -30 | -8 |
| 25 | 10 | 25 | -25 | -8 |
| 30 | 15 | 25 | -25 | -8 |
| 25 | 25 | 26 | -25 | -8 |
| 60 | 25 | 8 | -20 | -9 |
| 10 | 0 | 30 | -20 | 10 |
| 10 | 0 | 0 | -20 | 4 |
| 10 | 0 | 0 | -15 | 4 |
| 10 | 0 | 0 | -10 | -4 |
| 0 | 0 | 14 | -10 | -4 |
| 0 | 0 | 14 | -15 | -10 |
| 0 | 0 | 14 | -20 | -14 |
| 0 | 25 | 14 | -20 | -14 |

Table 1 Different situations used in the neural network

Where

LOD= Left Obstacle distance

ROD= Right Obstacle distance

FOD= Front Obstacle distance

TA=Target Angle

SA=Steering angle

CHAPTER 4

# ALGORTHIMS FOR PROGRAMMING IN MATLAB

## Algorithm for training neural network-:

1. Run MATLAB

2. The input matrix and the target matrix were defined.

3. Create feed-forward back-propagation network using the command [30] net = newff([S1 S2...S(N-l)],{TF1,TF2...TFNl}); where, Si size of ith layer, for N-1 layers, TFi Transfer function of ith layer. (Default = 'tansig' for hidden layers and 'purelin' for output layer. Various transfer functions [30] are as follows

Compet: Competitive transfer function

Hardlim: Hard limit transfer function

Hardlims: Symmetric hard limit transfer function

Logsig: Log-sigmoid transfer function

Netinv:Inverse transfer function

Poslin: Positive linear transfer function

Purelin: Linear transfer function

Radbas: Radial basis transfer function

Satlin: Saturating linear transfer function

Satlins: Symmetric saturating linear transfer function

Softmax: Softmax transfer function

Tansig:Hyperbolic tangent sigmoid transfer function

Tribas: Triangular basis transfer function4. The numbers of hidden layers were varied to get the best possible result by hit and trail.

4. Set the train function using different commands as follows. There are many train functions [30] such as

a. trainb: Batch training with weight and bias learning rules

b. trainbfg: BFGS quasi-Newton backpropagation

c. trainbfgc: BFGS quasi-Newton backpropagation for use with NN model reference adaptive controller

d. trainbr: Bayesian regularization

e. trainbuwb: Batch unsupervised weight/bias training

f. trainc: Cyclical order incremental update

g. traincgb: Powell-Beale conjugate gradient backpropagation

h. traincgf: Fletcher-Powell conjugate gradient backpropagation

i. traincgp: Polak-Ribiére conjugate gradient backpropagation

j. traingd: Gradient descent backpropagation

k. traingda: Gradient descent with adaptive learning rule backpropagation

l. traingdm: Gradient descent with momentum backpropagation

m. traingdx: Gradient descent with momentum and adaptive learning rule backpropagation

n. trainlm: Levenberg-Marquardt backpropagation

o. trainoss: One step secant backpropagation

p. trainr: Random order incremental training with learning functions

q. trainrp: Resilient backpropagation (Rprop)

r. trains: Sequential order incremental training with learning functions

s. trainscg: Scaled conjugate gradient backpropagation

5. Set the train parameters using the commands [30] below

net.trainParam.epochs -:Maximum number of epochs to train

net.trainParam.goal-: Learning rate

net.trainParam.max_fail-: Maximum validation failures

net.trainParam.mc-: Momentum constant

net.trainParam.min_grad-: Minimum performance gradient

net.trainParam.show -:Epochs between showing progress

net.trainParam.show-:CommandLine Generate command-line output

net.trainParam.show-:Window Show training GUI

net.trainParam.time -:Maximum time to train in seconds automotive.

6. Train the network.using the command "net = train(net,input,target)";

7. Simulate the network for finding the plot for Mean square Error, Gradient, Learning Rate Increment and Regression plot by using the command output=sim(net,input)

CHAPTER 5

# RESULTS AND DISCUSSION

# Results and Discussion

From the analysis of Neural Networks the navigation mechanism for an underwater robot has been developed.

The Feed forward back propagation algorithm is used to determine the steering angle for the underwater robot.

The neural network used for the navigation control of the underwater robot is provided with four inputs: Left Distance (Distance between the underwater robot and the nearest obstacle to its left); Right Distance (Distance between the underwater robot and the nearest obstacle to its right); Front Distance (Distance between the underwater robot and the nearest obstacle ahead of it) and Heading Angle (position of the underwater robot with respect to the target expressed as an angle). One output was generated for navigation of the underwater robot: Steering Angle (The angle to which the underwater robot must be turned)

After the underwater robot has been trained with the set of inputs, it is expected that the robot steers by itself without any human involvement in a nonlinear and unpredictable environment. This steering of the underwater robot is controlled by the outputs generated by the neural network. The steering angle obtained matches with the angle initially desired. This desired output angle for the particular set of obstacle distances had been fixed by a human. Thus the underwater robot should be capable of performing steering control automatically replacing human involvement.

The Feed forward neural network had been created using the MATLAB software.

The following results have been obtained after simulating the neural network using MATLAB.

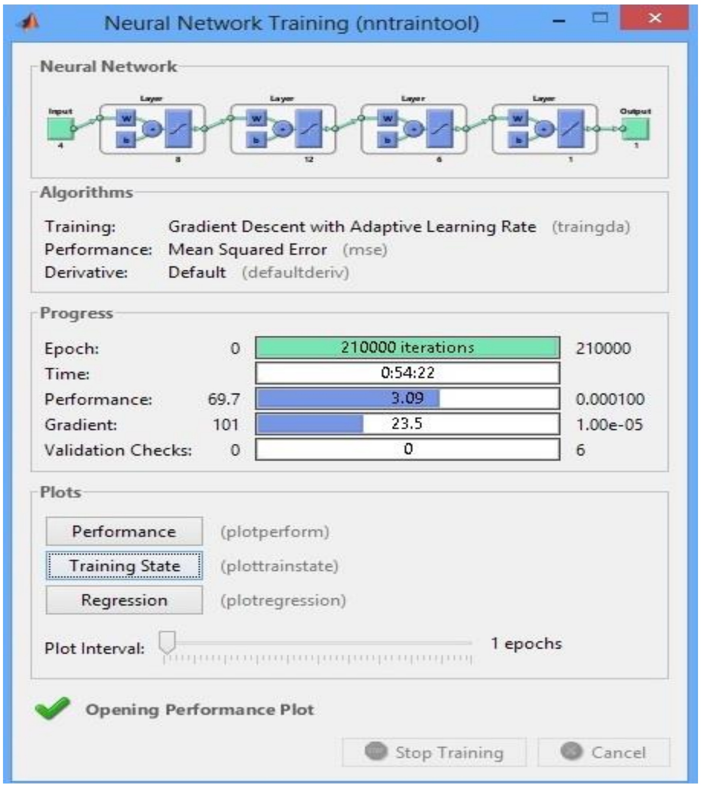


Figure-5 Neural Network Training in MATLAB

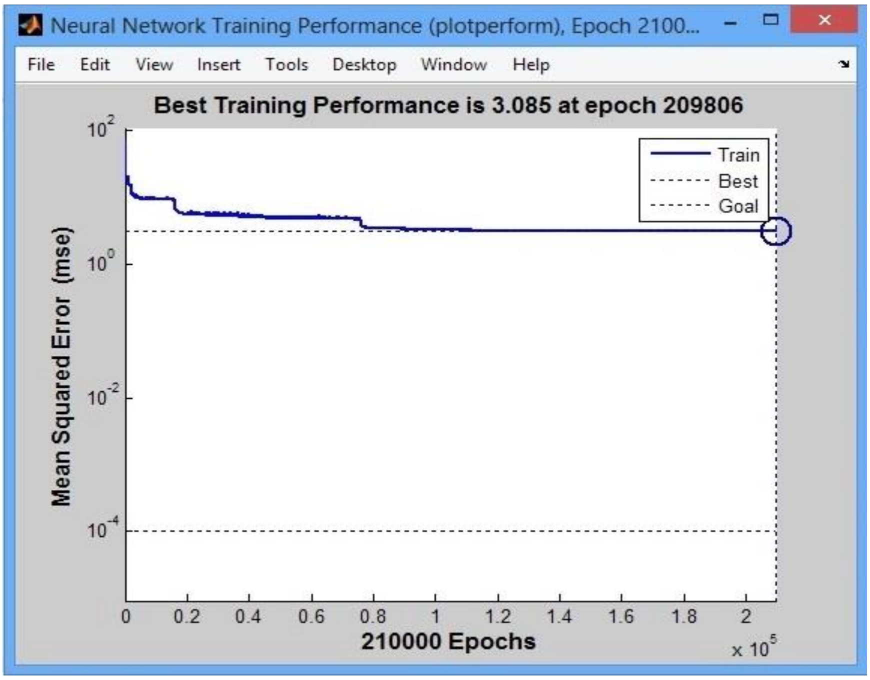


Figure 6-Variation of Mean Square Error with no of epochs

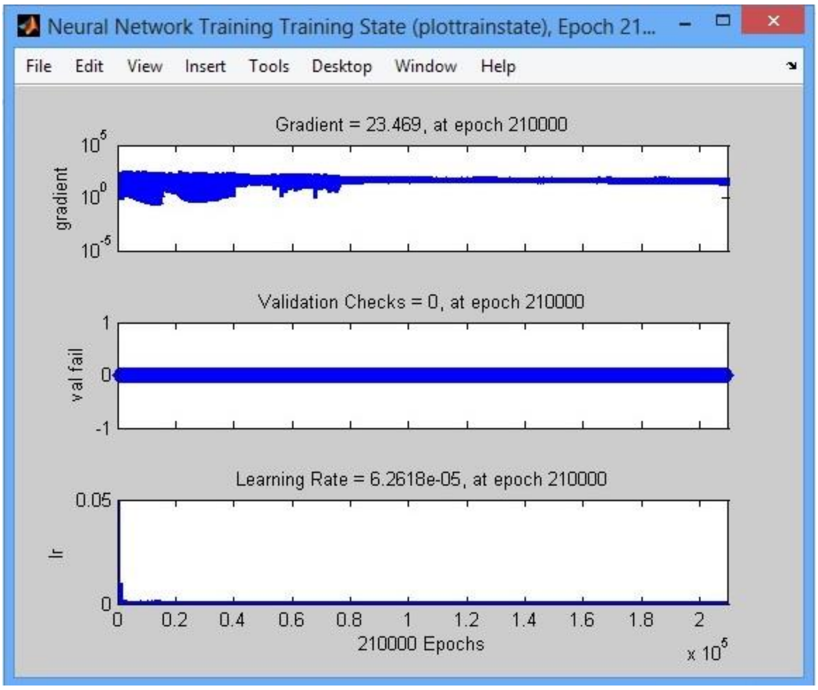


Figure 7-Variation of Gradient and Learning Rate

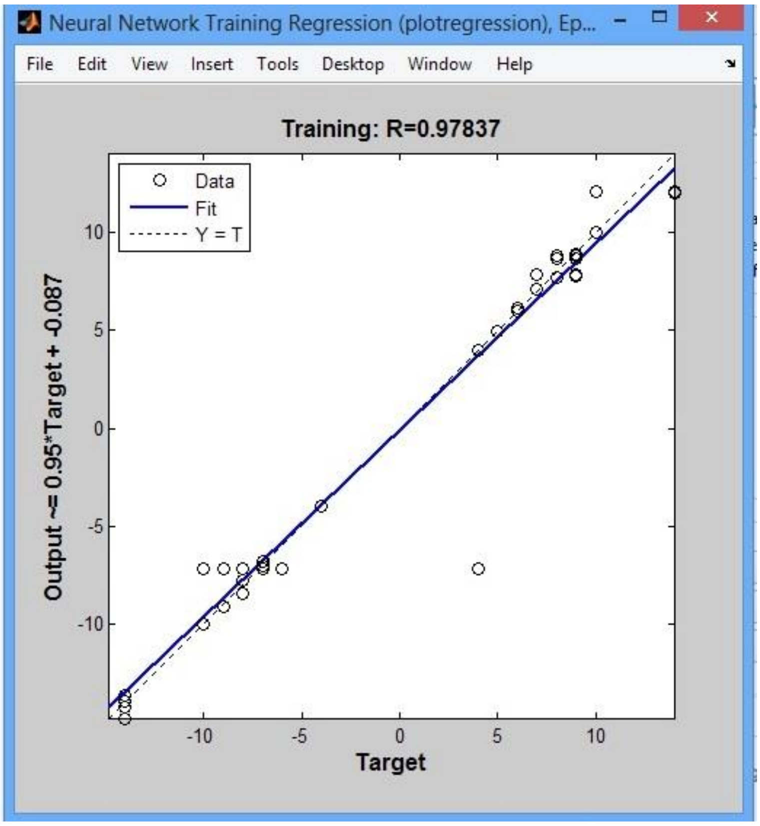


Figure 8- Regression plot of the neural network

The Mean Square Error, gradient, and the regression obtained was 3.085, 23.469, 0.97837 respectively.

Since the mean square error obtained is too less we can use this neural network for the navigation control of the underwater robot.

This trained neural network is fed to the virtual robot which is created using MATLAB. We placed some obstacle between the robot and the target. The path followed by the robot is as follows. The following simulation is obtained using MATLAB.

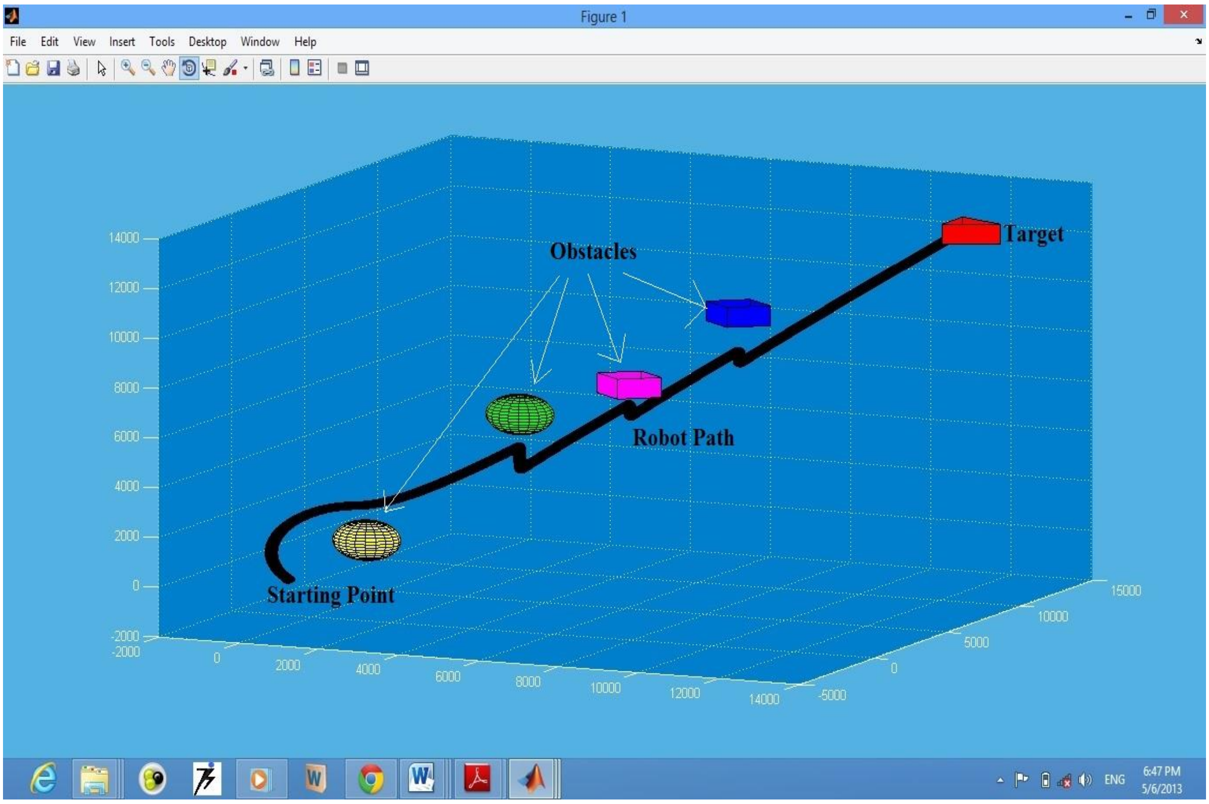


Figure 9-Simulation of Underwater robot with obstacles

CHAPTER 6

# CONCLUSION AND SCOPE FOR FUTURE WORK

# Conclusion and Scope for future works

A simulation of navigation control of underwater robot has been developed using MATLAB. The trained neural network was fed to the simulation of the underwater robot. Since trained neural network have a less Mean square error the simulation performance of the underwater robot is accurate. The simulation results show that the trained neural network can be used for an underwater robot which can be used for an unpredictable a non-linear environment. The performance of the trained neural network is compared with other models and they show an good agreement. The established neural network used in the robot has the following characteristics

1. They can avoid any obstacles which come along its path.

2. It has got a less mean square error which results in smooth and accurate navigation control of the underwater robot under any unpredictable environment.

3. The environment is identified by the underwater robot which gives sufficient data for path optimizing the path while navigating.

Further, the neural network model can be optimized by training with increased set of inputs. Since the simulation shows a good navigation result the neural network designed can be used practically in an underwater robot.

## References


1. Chen M., Jiang B. , Zou J. and Feng X., "Robust Adaptive Tracking Control of the Underwater Robot with Input Nonlinearity Using Neural Networks," International Journal of Computational Intelligence Systems, Vol.3, No. 5 ,October, 2010, 646-655.
2. Parhi D. R. and Singh M.K., "Intelligent Neuro-Controller for Navigation of Mobile Robot," International Conference on Advances in Computing, Communication and Control (ICAC3), 2009, 123-128.
3. Guo S., Gao B. and Xiao N., "Path-planning of Underwater Microrobot in 3-D Space Using Spiral Particle Pathway Searching Approach," Proceedings of the IEEE International Conference on Automation and Logistics Shenyang, China, August 2009, 1989-1994
4. Ishi K. and Ura T., "An adaptive neural-net controller system for an underwater vehicle," Control Engineering Practice 8, 2000, 177-184.
5. Zhang M. and Chu Z., "Adaptive sliding mode control based on local recurrent neural networks for underwater robot" Ocean Engineering 45, 2012, 56–62.
6. van de Ven P.W.J., Johansen T.A., Sørensen A.J., Flanagan C. and Toal D., "Neural network augmented identification of underwater vehicle models," Engineering Applications of Artificial Intelligence 18, 2005, 533–547.
7. Kim T.W., Yuh J., "Application of on-line neuro-fuzzy controller to AUVs," Information Sciences 145, 2002, 169–182
8. Lianzhi Y., Weichong L., "Cascade Hopfield Neural Network Model and Application in Robot Moving Process" Procedia Engineering 29, 2012, 814 – 818
9. Joshi M.M. and Zaveri M.A, "Optimally Learnt, Neural Network Based Autonomous Mobile Robot Navigation System," ACEEE Int. J. on Control System and Instrumentation, Vol. 02, No. 01, Feb 2011, 28-32.
10. Lin L., Xie H., Zhang D. and Shen L., "Supervised Neural Q_learning based Motion Control for Bionic Underwater Robots," Journal of Bionic Engineering 7 Suppl., 2010, S177–S184.
11. Pradhan S.K., Parhi D.R. and Panda A.K., "Navigation of Multiple Mobile Robots usingRule-based-Neuro-Fuzzy Technique," International Journal of Computational Intelligence Spring 2007 142-152

12. Kundu S., Parhi D.K. and Deepak B.B.V.L, “Fuzzy-Neuro based Navigational Strategy for Mobile Robot,” International Journal of Scientific & Engineering Research, Volume 3, Issue 6, June-2012I,SSN 2229-5518.
13. Kodogiannis V.S., Lisboa P.J.G. and Lucas J., “Neural Network Based Predictive Control Systems For Underwater Robotic Vehicles,” Department of Electrical Engineering and Electronics, University of Liverpool, UK, 369-376.
14. Fernandes J.M.M., Tanaka M.C., Freire Jr. R.C.S., and Bessa W.M., “A Neural Network Based Controller For Underwater Robotic Vehicles,” ABCM Symposium Series in Mechatronics - Vol. 5, Section II – Control Systems, 2012, 456-464.
15. Bagheri A. and Moghaddam J.J., “Simulation and tracking control based on neural-network strategy and sliding-mode control for underwater remotely operated vehicle,” Neurocomputing 72, 2009, 1934–1950.
16. Bagheri A., Karimi T. and Amanifard N., “Tracking performance control of a cable communicated underwater vehicle using adaptive neural network controllers,” Applied Soft Computing 10, 2010, 908–918.
17. Chatchanayuenyong T. and Parnichkun M., “Neural network based-time optimal sliding mode control for an autonomous underwater robot,” Elsevier, Mechatronics 16, 2006, 471–478.
18. Fujii T. and Ura T., “SONCS: Self-organizing Neural-Net-Controller System for Autonomous Underwater Robots,” Institute of Industrial Science, The University of Tokyo. CH 3065-0/91/0000-1973,IEEE, 1973-1982
19. Li J.H., Lee P.M., and Jun B.H., “A Neural Network Adaptive Controller for Autonomous Diving Control of an Autonomous Underwater Vehicle,” International Journal of Control, Automation, and Systems, vol. 2, no. 3, September 2004, 374-383.
20. Hoang N.Q., Kreuzer E., “Adaptive PD-controller for positioning of a remotely operated vehicle close to an underwater structure: Theory and experiments,” Control Engineering Practice 15, 2007, 411–419
21. Lee M. and Choi H.S., “A Robust Neural Controller for Underwater Robot Manipulators” Ieee Transactions On Neural Networks, Vol. 11, No. 6, November 2000, 1465-1470.
22. Ouarda H., “A Neural Network Based Navigation for Intelligent Autonomous Mobile Robots” International Journal Of Mathematical Models And Methods In Applied Sciences, Issue 3, Volume 4, 2010, 177- 186.

23. Zhang D., Hu D., Shen L. and Xie H., “Design of an artificial bionic neural network to control fish-robot’s locomotion” Elsevier, Neurocomputing 71, 2008, 648–654.

24. Birsel A., Billur B.; "Neural networks for improved target differentiation and localization with sonar”; Neural Networks 14, 2001, 355-373.

25. Mustafa S., Ozdemirel N.E., Sinan K.; ―Theory and Methodology A self-organizing neural network approach for the single AGV routing problem‖; European Journal of Operational Research 121, 2000, 124-137.

26. Orille A. L., Sowilam G.M.; ―An Application Of Neural Networks For Direct Torque Control‖; Computers & Industrial Engineering 37, 1999, 391-394.

27. Franco S., Chung T.A.; ―Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results‖; Neural Networks,Vol. 11, No. 1, 1998, 15–37

28. Christos S. and Dimitrios S.; Neural Networks; http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Architecture of neural networks

29. Rodríguez F. J., Neural Network Tutorial; http://www.gc.ssr.upm.es/inves/neural/ann1/anntutor.htm

30. Matlab Central
http://www.mathworks.in/matlabcentral/